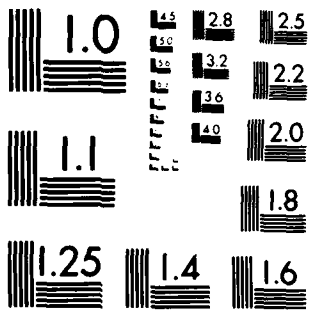
AD-A141 444 A COMPARATIVE STUDY OF LINEAR ARRAY SYNTHESIS TECHNIQUE USING A PERSONAL COMPUTER(U) NAVAL RESEARCH LAB WASHINGTON DC S R LAXPATI ET AL. 29 MAY 84 NRL-MR-5336 1/1

UNCLASSIFIED F/G 9/2 NL

END
DATE FILMED
7-84
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AD-A141 444

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): NRL Memorandum Report 5336

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION: Naval Research Laboratory

6b. OFFICE SYMBOL (If applicable): Code 5370

6c. ADDRESS (City, State and ZIP Code): Washington, DC 20375

7a. NAME OF MONITORING ORGANIZATION

7b. ADDRESS (City, State and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Naval Electronic Systems Command

8b. OFFICE SYMBOL (If applicable)

8c. ADDRESS (City, State and ZIP Code): Washington, DC 20360

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
62712N		XF12-141-100	53-1854-A4

11. TITLE (Include Security Classification): (See page ii)

12. PERSONAL AUTHOR(S): S.R. Laxpati, J.P. Shelton,* and M.A. Burns**

13a. TYPE OF REPORT: Interim

13b. TIME COVERED: FROM TO

14. DATE OF REPORT (Yr., Mo., Day): May 29, 1984

15. PAGE COUNT: 26

16. SUPPLEMENTARY NOTATION: *Symmetron, Inc., Arlington, VA 22201 **Voice of America, Washington, DC 20547

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Array synthesis; Chebyshev array; Taylor array computer codes

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Five computer programs for synthesizing low-sidelobe sum patterns from linear arrays are evaluated in terms of run time and precision. Three of the programs are based on the Dolph-Chebyshev synthesis procedure, in which all sidelobes are set at the same level. The other two programs are based on a discretized version of the Taylor synthesis procedure, in which far-out sidelobes are allowed to decay. The programs were written for use on small 8- and 16-bit personal computers. It was found that the fastest running programs are also the most precise. The only Chebyshev program that gave satisfactory precision for arrays as large as 100 elements is based on Bresler's nested product algorithm, and the only similarly acceptable Taylor program is based on Shelton's discretized synthesis.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED X SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: S. R. Laxpati

22b. TELEPHONE NUMBER (Include Area Code): (202) 767-6277

22c. OFFICE SYMBOL: Code 5370

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE

11. TITLE (Include Security Classification)

A COMPARATIVE STUDY OF LINEAR ARRAY SYNTHESIS TECHNIQUE USING A PERSONAL COMPUTER

CONTENTS

A COMPARATIVE STUDY OF LINEAR ARRAY SYNTHESIS TECHNIQUE USING A PERSONAL COMPUTER

INTRODUCTION

Procedures for synthesizing the radiation patterns of linear arrays based on the specification of its sidelobe structure are well established. One of these is the technique originally proposed by Dolph[1] and it provides for a uniform sidelobe level. Another technique, although developed for line sources, is due to Taylor[2] and can be adopted for linear arrays. The latter is popular due to its synthesized aperture distributions which are more readily realizable. A discrete version of the Taylor synthesis procedure is discussed by Shelton[3].

Both Dolph-Chebyshev and Taylor synthesis techniques, fundamentally, rely on manipulation of the zeros of the linear array pattern function. The aperture distribution for the desired pattern function usually requires lengthy computations. In case of Dolph-Chebyshev synthesis, this problem has been addressed by several authors[4-9] over the past few decades. The Taylor synthesis, in effect, uses a discrete Fourier Transform technique (called Woodward[10] synthesis) to obtain the aperture distribution. These procedures do not have much in common; as a matter of fact, in case of an endfire Chebyshev array, expression for the element excitations are quite different from that for a broadside Chebyshev array. However, the knowledge of the pattern null locations in the above synthesis procedures can be used to develop a simple expression that is suitable for all cases. The expression is readily developed based on the convolution synthesis procedure discussed by Laxpati[11] for planar arrays.

With several alternate expressions being available for the aperture distribution of Dolph-Chebyshev and Taylor syntheses, it is desirable to undertake a study to make some recommendations as to the suitability of these expressions in numerical computation. Due to the increasing use of personal computers by antenna engineers, it is felt that an investigation of this nature should be confined to the computation using such small computers. Thus, in this paper, we present the results of a comparative study of various linear array synthesis techniques. In the next section, after a brief discussion of the three basic techniques for evaluation of Chebyshev coefficients, we discuss the accuracy and computation times associated with these techniques. The following section presents the results of the study involving two different techniques (one due to Shelton[3] and the other using the convolution procedure) for Taylor synthesis. In the last section, some general observations about the investigation and on the results are offered.

DOLPH-CHEBYSHEV SYNTHESIS

Following Dolph's paper on Chebyshev synthesis, Barbiere[4], Van Der Maas[5], Salzev[6], and Brown[7,8] reported on alternative means of evaluating aperture distribution for Chebyshev arrays. Although they are not the same, the expressions by Barbiere, Salzev and Brown are

Manuscript approved February 21, 1984.

similar in that they express the current in an element in terms of a finite series of terms involving ratios of factorial functions and with alternating sign. The expression by Elliott[12] is representative of this group and is the one used in this work and is reproduced below. We shall call this the classical expression. Also, although our results are valid for odd or even number of elements, for simplicity, we will present examples of odd number of elements. Thus, all linear arrays discussed in the following have (2N+1) elements; the element numbering scheme is shown in figure 1, where the elements are assumed to have a symmetric excitation leading to the broadside radiation.

Classical Technique:

$$I_n = \sum_{p=n}^{N} (-1)^{N-p} \frac{N}{N+p} \frac{\Gamma(N+p+1)}{\Gamma(N-p+1)\Gamma(p+n+1)\Gamma(p-n+1)} (u_o)^{2p} \qquad (1)$$

$$n=0,1,2,\ldots,N.$$

where $T_{2N}(u_o)=R$ and SLL = 20 log R. Here, SLL is the desired sidelobe level in dB, $T_{2N}(x)$ is the Chebyshev polynomial of degree 2N and $\Gamma(x)$ is the Gamma function.

In contrast, the expression given by Van Der Maas involves terms of the same sign inside the summation. Bresler[9] reformulated the expression into a recursive form using nested products. This, we feel is a distinctly different form of representation of the coefficients. Thus, we use this representation (called Nested Product Technique) in our comparison. This expression (in our notation) is shown below.

Nested Product Technique:

$$I_{N-n} = 2N\ \alpha\ NP(n,f_m,\alpha). \quad n=0,1,2, \qquad (2)$$

where $NP(n,f_m,\alpha) = \sum_{m=1}^{n} \alpha^{n-m} \prod_{j=m}^{n} f_j;\ f_n \equiv 1.$

and $f_m = \frac{m(2N-2n+m)}{(n-m)(n+1-m)}$;

also $\alpha = 1 - \frac{1}{u_o^2}$.

The third technique is based on the convolution of three element canonical arrays[11]. These canonical arrays have outer element excitations of unity, whereas the center element excitation c_j; for j=1,2, ...N is chosen such that the j^{th} canonical array has a pattern null at the location of the j^{th} symmetric zero pair of the Chebyshev

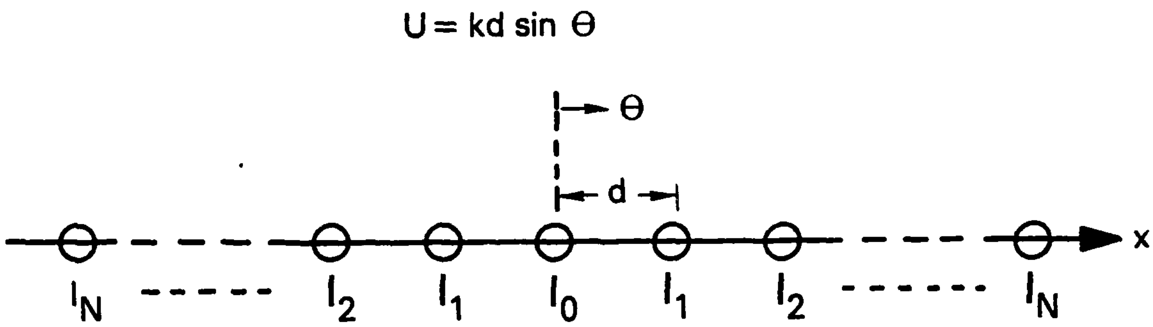


Figure 1 - (2N + 1) Element Linear Array

polynomial. These arrays are then convolved to generate the large array. Convolution Technique:

$$w_{oj} = \cos \frac{(2j-1)}{2N} \pi \ ; \ j=1,2,...,N.$$

Where w_{oj} are the zeros of the Chebyshev polynomial $T_{2N}(w)$.

$$C_j = -2\cos u_{oj};$$

$$u_{oj} = 2\arccos (w_{oj}/u_o).$$

And the aperture distribution

$$I(x) = \sum_{n=-N}^{N} I_n \delta(x-nd) = f_1 * f_2 * --- * f_N \qquad (3)$$

where $f_j = \delta(x-d) + C_j\delta(x) + \delta(x+d)$.

Using these three expressions (equations (1), (2) and (3)), computer programs NESTED, CHEB and CONCHEB, respectively, were written to implement the Chebyshev synthesis. Different versions of the program suitable for implementation on different machines were written. These were two personal computers used in the numerical phase; one is an 8-bit Radio Shack TRS-80 Model II which has available an interpretive BASIC language. The other computer is a 16-bit NEC Advanced Personal Computer with BASIC and FORTRAN IV compilers. Also, in order to ascertain the numerical accuracy, some of the programs were run on a 32-bit mainframe computer (Texas Instrument's Advanced Scientific Computer at the Naval Research Laboratory) using double precision (REAL*8) arithmetic.

The computation was carried out for several different array sizes ranging from 15 to 99 elements; although, in principle, there is no limit to the size of arrays that may be synthesized. Furthermore, all designs specified a sidelobe level of 30 dB.

Figure 2 shows the run time, under FORTRAN, for the three aforementioned Chebyshev synthesis programs versus number of elements. The CHEB program was the slowest; but more importantly, the program failed to converge to the correct element excitations beyond 30 elements. Over 21 elements the accuracy of the excitation was only to 2 digits. When the program was run using double precision arithmetic it still failed to converge above 31 elements. This indicates that the classical technique inherently has a limitation as to the largest size of array that may be synthesized.

The convolution synthesis program, CONCHEB, although much faster than CHEB, certainly cannot complete with NESTED program in speed. Also, beyond 61 elements, the CONCHEB program failed to converge. The current version of the program convolves three-element arrays using zeros of the Chebyshev polynomial in an alternating sequence; i.e., the

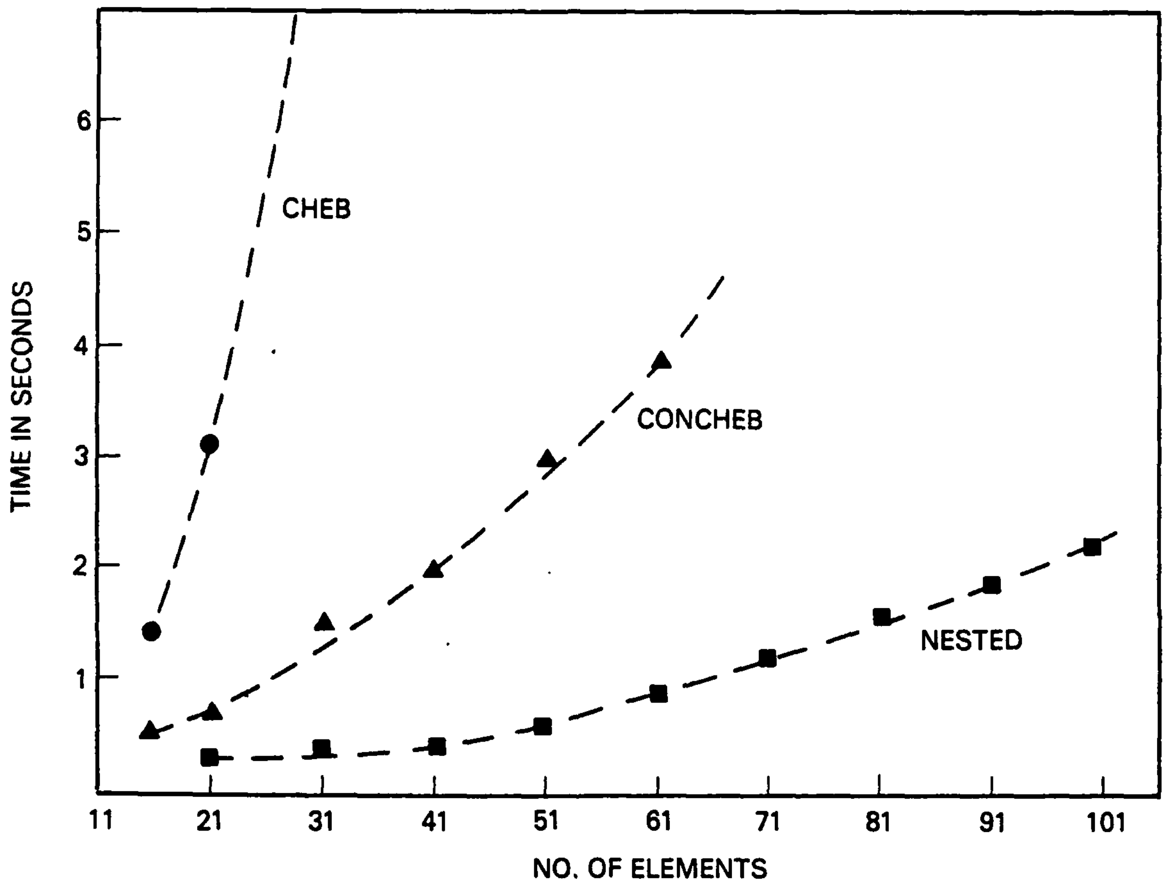


Figure 2 - Run Times for Chebyshev Programs -
FORTRAN IV on NEC-APC

sequence in j is 1,N,2,N-1,3,---. No attempt was made to modify this convolution process to improve the accuracy; past experience with the convolution process indicates that some improvement may be possible. However, with reference to figure 2, it is obvious that the NESTED program is the most efficient one.

The results of the element excitations indicate that it is an extremely stable algorithm; provides a very good accuracy in single precision (six digit accuracy); and of course, it is very fast. This program, NESTED, was translated into BASIC and run on the TRS-80, Model II computer. The execution time ranged from two seconds for 21 element array to 36 seconds for a 99 element array. Although, the execution times in BASIC are about 15 to 20 times longer than that in FORTRAN, they are not significantly long to be of any major consequence. The NESTED program was also run using double precision (16 significant digits) on the mainframe computer. The total execution time for all 10 different arrays was less than 0.3 seconds!

Our experience with synthesis of various Chebyshev arrays using these three different techniques clearly demonstrates that the most important consideration on small computers is not the speed of execution but the accuracy of the final result. In this sense as well, the nested product algorithm proposed by Bresler[9] is the winner.

TAYLOR SYNTHESIS

Synthesis procedure proposed by Taylor[2] applies to a continuous aperture. In practice, this procedure is used for discrete aperture (arrays) by properly discretizing the continuous distribution. Shelton[3] presented a synthesis procedure for discrete aperture distribution for Taylor type sidelobe structure. He expressed the pattern function in the form of a product function of zeros and then carried out the synthesis exactly analogous to that by Taylor; that is, to use the Woodward synthesis technique. In particular, for a 2N+1 element array, all 2N zeros are explicitly specified in the pattern function. Thus, analogous to the Chebyshev synthesis, this synthesis is amenable to the convolution procedure. In view of this, in the case of Taylor synthesis, we compare the two techniques; one proposed by Shelton and the other being the convolution synthesis. Before presenting and discussing the results of the investigation, the pertinent expressions for the two syntheses are given below. Once again, we will limit out discussion to arrays with odd (2N+1) number of elements.

Discrete Taylor (Shelton[3]) Technique:

$$u_{on} = \frac{2\pi\bar{n}}{(2N+1)} \frac{\sqrt{A^2+(n-1/2)^2}}{\sqrt{A^2+(\bar{n}-1/2)^2}} \quad , \; n=1,2,---,\bar{n}-1$$

$$= \frac{2\pi n}{(2N+1)} \quad , \; n=\bar{n},---,N. \tag{4}$$

where $A = \frac{1}{\pi}\cosh^{-1}(R)$; $\bar{n}$ is equal to the number of near-in zeros that are moved in order to achieve the desired sidelobe ratio R (or equivalently the number of near-in sidelobes that are required at the specified level). The element excitations are

$$I_p = 1+2\sum_{m=1}^{\bar{n}-1} a_m \cos\frac{2mp\pi}{2N+1} \quad , \; p=0,---,N \; . \tag{5}$$

where

$$a_m = E\left(\frac{2\pi m}{2N+1}\right) ;$$

$$E(u) = \prod_{n=1}^{N} \frac{(\cos u-\cos u_{on})}{(1-\cos u_{on})} \; .$$

For the case of the convolution synthesis procedure, once the symmetric zero pairs are established, the excitation of the center element of a three element canonical array is readily determined. The procedure and expressions are analogous to the case of Chebyshev convolution synthesis. They are

zeros are $\pm u_{oj}$; $j=1,2,---,N$

where u_{oj} are defined through equation (4), and the excitation

$C_j = -2\cos u_{oj}$.

The synthesis of the large array is carried out using the convolution of three element arrays, chosen in the same alternating zero sequence as indicated for the Chebyshev array.

Based on these two procedures, computer codes STAYL and CONTAYL, respectively, were developed in FORTRAN using single precision arithmetic Run time associated with these codes for $\bar{n}$ = 6 and the sidelobe level of 30 dB for various number of elements from 15 to 99 were recorded and are shown in figure 3.

The program CONTAYL failed to converge, once again, for arrays with more than 71 elements and provided only two to three digit accuracy between 31 and 61 elements. These results are similar to the Chebyshev convolution synthesis. Even the run time data is very close.

The computation time associated with STAYL has an interesting behavior with increasing number of elements; it is almost linear. This is to be expected, since the number of computations to be carried out for each element is determined by $\bar{n}$ and not (2N+1). The corresponding growth for CONTAYL is exponential. Thus, for small number of elements CONTAYL may save some computation time but will suffer in accuracy as

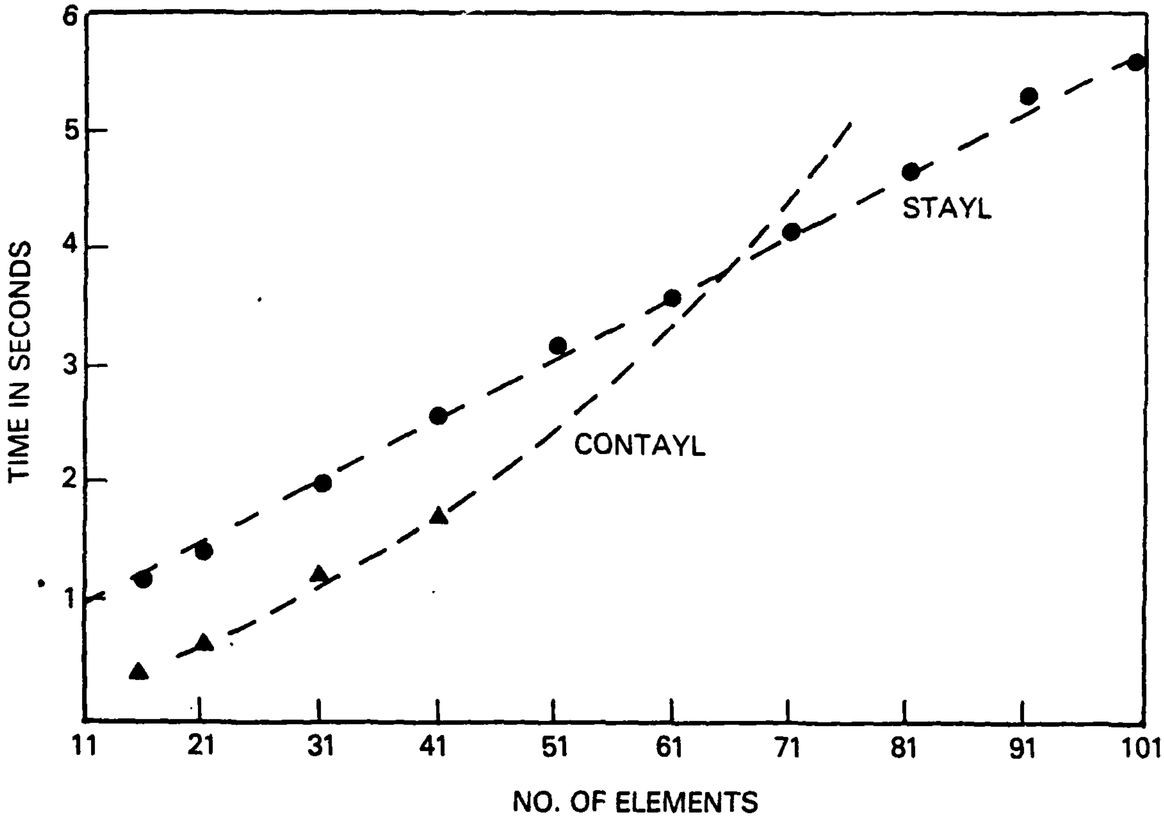


Figure 3 - Run Times for Taylor Programs -
FORTRAN IV on NEC-APC

the number of elements increases. A check of STAYL program using double precision arithmetic on the mainframe computer indicates that it has five to six digit accuracy in single precision on a small computer.

It should be noted that the STAYL program code was developed by Shelton for the HP-41C, a pocket calculator. On this calculator, one has 10 significant digit capacity and thus the results obtained are more accurate than with a single precision FORTRAN Code. But, as one would expect, the HP-41C is very slow; it took approximately 5 minutes to synthesize a 31 element array.

STAYL Code was also run on NEC-APC using CBASIC, a compiler BASIC. In CBASIC, the computation times were significantly higher, ranging from 30 seconds for a 15 element array to 217 seconds for a 99 element array. However, the computation was carried out to 14 significant figures.

Once again, as with Chebyshev synthesis, we find the overriding consideration in Taylor synthesis is not the computation time, but the accuracy of the results. In this sense, Shelton's procedure is most efficient.

CONCLUSIONS

As is often the case with engineering investigations, the most significant results presented in this paper are not what we were looking for when we began the project. We were originally interested in evaluating computer run times for the various programs. However, two points soon became apparent -- first, most of the programs run fast enough, even on small machines, so that run time is not a major concern, and second, only two of the programs give adequate precision for the range of array size that was investigated. It is concluded that Bresler's nested product algorithm gives excellent results in terms of speed and precision, and also that Shelton's discretized procedure allows precise Taylor synthesis for all sizes of arrays. Finally, it is noted that the programs are very brief; the FORTRAN computer codes for all five programs are included in the appendix and the codes in BASIC are also available from the authors.

REFERENCES

1. C.L. Dolph, "A Current Distribution for Broadside Arrays Which Optimizes the Relationship Between Beam Width and Side-Lobe Level", Proc. IRE, 34 , No.6, pp. 335-348, June 1946.

2. T.T. Taylor, "Design of Line-Source Antennas for Narrow Beamwidth and Low Side Lobes", IRE Tran. Antennas and Propagat., AP-3, pp. 16-28, No. 1, January 1955

3. J.P. Shelton, "Synthesis of Taylor and Bayliss Patterns for Linear Antenna Arrays", NRL Report 8511, Naval Research Laboratory, Washington, D.C. 20375, August 31, 1981. (AD-A103 534)

4. D. Barbiere, "A Method of Calculating the Current Distribution of Tschebyscheff Arrays", Proc. IRE, 40, No. 1, pp. 78-82, January 1952.

5. G.J. Van Der Maas, "A Simplified Calculation for Dolph-Tchebycheff Arrays", J. App. Phys., 25, No. 1, pp. 121-124, January 1954.

6. H.E. Salzer, "Note on the Fourier Coefficients for Chebyshev Patterns", Proc. IEE (London), 103C, pp. 286-288, February 1956.

7. J.L. Brown, Jr., "A Simplified Derivation of the Fourier Coefficients for Chebyshev Patterns", Proc. IEE (London), 105C, pp. 167-168, November 1957.

8. J.L. Brown, Jr., "On the Determination of Excitation Coefficients for a Tchebycheff Pattern", IRE. Tran. Antennas and Propagat, AP-10, pp. 215-216, March 1962.

9. A.D. Bresler, "A New Algorithm for Calculating the Current Distributions of Dolph-Chebyshev Arrays", IEEE Tran. Antennas and Propagat., AP-28, No. 6, November 1980. pp. 951-952.

10. P.M. Woodward, "A Method of Calculating the Field Over A Plane Aperture Required to Produce A Given Polar Diagram", Journal of IEE (London), Pt. III A, 93, pp. 1554-1558, 1946.

11. Sharad R. Laxpati, "Planar Array Synthesis with Prescribed Pattern Nulls", IEEE Trans. on Antennas and Propagat., AP-30, No. 6, pp. 1176-1183, November 1982.

12. R.S. Elliott, "Antenna Theory and Design", Prentice-Hall, Inc., pp. 143-147, 1981.

APPENDIX

In this appendix the FORTRAN IV computer codes for NEC-APC with supersoft FORTRAN compiler are listed. As noted in the main body of the report, the programs are brief; there are a number of "comment" statements in the listing and thus are easy to follow. No sample inputs or outputs are included.

Programs listed in the following are NESTED, CHEB, CONCHEB, STAYL and CONTAYL.

```
C                          PROGRAM NESTED
C  REVISED 01/28/84
C  CHEBYSHEV ARRAY ALGORITHM USING NESTED PRODUCTS FORMULATION
C  INPUT IS M = NO. OF ELEMENTS; SLL = SIDELOBE LEVEL IN DB.
      REAL I(100), NP, C(100)
      WRITE (1,100)
  100 FORMAT (' ENTER DATA: M,SLL')
      READ (1,200) M,SLL
  200 FORMAT ( I0,F0.0)
      N = M / 2
      TEST = (-1)**M
      IF(TEST.GT.0) GO TO 10
      N = (M - 1) / 2
   10 R = 10. ** ( SLL / 20. )
      ARCOSH = ALOG ( R + SQRT ( R**2. - 1 ) )
      A = ARCOSH / (M-1)
      ALPHA = ( TANH (A) ) ** 2.
      I(N+1) = 1.0
      I(N) = (M-1) * ALPHA
      DO 30  K = 2, N
             NP = 1.0
             DO 20  J = 1, K-1
                    FN = J * (M-1-2*K+J)
                    FD = (K-J) * (K+1-J)
                    F = FN/FD
                    NP = NP * ALPHA * F + 1.
 20          CONTINUE
             I(N+1-K) = ( M-1 ) * ALPHA * NP
 30   CONTINUE
      DO 40  L = 1, N+1
             C(N+L) = I(L)
             C(N+2-L) = I(L)
 40   CONTINUE
      WRITE (4, 50)
 50   FORMAT ( '            CURRENTS')
      DO 60  L = 1, M
             WRITE (4,70) L, C(L)
 60   CONTINUE
 70   FORMAT (10X, I2, 10X, F10.6)
      STOP
      END
```

```
C                          PROGRAM CHEB
C REVISED 01/28/84
C BASED ON A CLASSIC METHOD OF COMPUTATION OF CHEBYSHEV
C EXCITATION VOLTAGES.
C REFERENCE ANTENNA THEORY AND DESIGN; ELLIOTT.
C ODD NUMBER OF ELEMENTS ONLY
      REAL CC(100),C(100),CRNT(100)
      WRITE (1,100)
100   FORMAT (' ENTER DATA: N, SLL')
      READ (1,200) N,SLL
200   FORMAT (I0,F0.0)
      M = N-1
      MM = M/2
      NN = (N+1)/2
      PI = 3.1415927
      R = 10. ** (SLL/20.0)
      U = COSH (RCOSH (R)/ FLOAT (M) )
      DO 10  I = 1,NN
            II = I -1
            C(I) = 0.0
        DO 20   J = I,NN
                JJ = J - 1
                A = FLOAT (NN + JJ)
                GA = GAMALN (NN + JJ)
                GB = GAMALN (NN - JJ)
                GE = GAMALN (J - II)
                GD = GAMALN (J + II)
                UP = U ** (2*JJ)
                SIGN = (-1) ** (NN-J)
                TL = EXP (GA - GB -GE - GD)
                TN = UP*SIGN*(2.*NN-1.)/(2.*A)
                T = TL * TN
                C(I) = C(I) + T
20              CONTINUE
10    CONTINUE
      DO 12 J = 1,NN
       CC(J) = C(J)/ C(NN)
12     CONTINUE
      DO 13 J = 1,NN
       CRNT (NN-1+J) = CC(J)
       CRNT (NN+1-J) = CC(J)
13     CONTINUE
      WRITE (4,30)
30    FORMAT ('          CURRENTS')
      DO 14 I = 1,N
       WRITE (4,300) I, CRNT(I)
14    CONTINUE
300   FORMAT (10X,I2,15X,F12.8)
      STOP
      END
```

```
C***********
C INVERSE HYPERBOLIC FUNCTION
C***********
      FUNCTION RCOSH (R)
      RCOSH = ALOG(R + SQRT(R*R - 1.0))
      RETURN
      END
C***********
C HYPERBOLIC FUNCTION
C***********
      FUNCTION COSH (R)
      Y = EXP (R)
      COSH = (Y + (1.0/Y) ) /2.
      RETURN
      END
C***********
C        GAMALN FUNCTION
C***********
      FUNCTION GAMALN (K)
      GAMALN = 0.0
      IF (K .EQ. 0) RETURN
      FACT = 0.0
      TPL = 0.91893853
      AL = K
10    IF (AL .GE. 10.0) GO TO 20
      FACT = FACT + ALOG (AL)
      AL = AL + 1.0
      GO TO 10
20    TERM = (AL - 0.5) * ALOG(AL) - AL + TPL
     1  + 1.0/(12.*AL) - 1.0/(360.0 * AL**3) + 1.0/(1260.*
     2  AL**5) - 1.0/ (1680. * AL**7)
      GAMALN = TERM - FACT
      RETURN
      END
```

```
                 PROGRAM CONCHEB
C*******
C*  LINEAR       ARRAY SYNTHESIS USING CONVOLUTION METHOD.
C*  CHEBYSHEV SIDELOBE DESIGN
C*  ODD NUMBER OF ELEMENTS.
C*******
      REAL PSI(100)
      REAL C(100),AA(100),A1(100),A2(100),A3(100),CONV(100)
      DATA A1/100*1./, A3/100*1./, CONV/100*0./, AA/100*0./
C*******
C*  N = NUMBER OF ELEMENTS IN THE ARRAY. MUST BE ODD!!
C*  SLL = SIDE LOBE LEVEL IN DBS.
C*******
      PI = 3.1415297
      WRITE (1,100)
100   FORMAT (' ENTER DATA: N,SLL')
      READ (1,200) N,SLL
200   FORMAT (I0,F0.0)
      M = N-1
      NN = (N+1)/2
      MM = (N-1)/2
      MD = (MM/2) + 1
20   CALL CHEBX(PI,M,NN,SLL,MM,PSI)
C********
C*  CREATE THREE ELEMENT ARRAYS
C********
30   DO 40 I = 1,MM
      A2(I)  =  -2. * COS(PSI(I))
40   CONTINUE
C********
C*  REPEATED CONVOLUTION OF 3-ELEMENT ARRAYS
C********
      CONV(1) = A1(1)
      CONV(2) = A2(1)
      CONV(3) = A3(1)
      L = 1
      K = 5
      LX = 0
50    LL = NN-L
      LX = LX+1
60    L = LL
      C(1) = A1(L)
      C(2) = A2(L)
      C(3) = A3(L)
      DO 70 I = 1,3
      DO 70 J = I,K
       JJ = J-I+1
       AA(J) = AA(J) + CONV(JJ)*C(I)
70    CONTINUE
      DO 80 I = 1,K
       CONV(I) = AA(I)
       AA(I) = 0.0
80    CONTINUE
      K = K+2
```

```
      IF (L.EQ.MD) GO TO 90
      LL = NN+1-L
      LSUM = L+LX
      IF (LSUM.EQ.NN) GO TO 60
      GO TO 50
90    CONTINUE
      WRITE (4,600)
600   FORMAT ( '         CURRENTS'/)
      DO 120 I = 1,N
       WRITE (4,700) I,CONV(I)
120   CONTINUE
700   FORMAT (10X,I2,10X,F10.6)
      STOP
      END
      FUNCTION COSH(R)
      Y = EXP(R)
      COSH = (Y + (1.0/Y))/2.
      RETURN
      END
C********
C*  INVERSE HYPERBOLIC COSINE FUNCTION
C********
      FUNCTION RCOSH(R)
      RCOSH = ALOG(R + SQRT(R*R - 1.0))
      RETURN
      END
C********
C*  CHEBYSHEV ZEROS
C********
      SUBROUTINE CHEBX(PI,M,NN,SLL,MM,PSI)
      REAL X(50),PSI(100)
      R = 10.0 **(SLL/20.)
      B = COSH(RCOSH(R)/M)
      DO 10 I = 1,NN
       J = I-1
       X(I) = COS(PI*(2.*J + 1.)/(2.*M))
10    CONTINUE
      DO 20 J = 1,MM
       II = NN-1+J
       JJ = NN-J
       Y = X(J) / B
       PSI(II) = 2.*ATAN(SQRT(1-Y*Y)/Y)
       PSI(JJ) = PSI(II)
 20   CONTINUE
      RETURN
      END
```

```
C                           PROGRAM STAYL
C REVISED 01/28/84
C THIS PROGRAM COMPUTES ELEMENT EXCITATIONS FOR TAYLOR
C TYPE SIDELOBES USING SYNTHESIS EXPRESSIONS OF SHELTON
      DIMENSION Z(100),AM(100),EN(100),EX(100)
      WRITE (1,100)
100   FORMAT (' ENTER N,NBAR,SIDELOBE LEVEL FOR TAYLOR'
     1  ,' SYNTHESIS' )
      READ (1,200) N, NBAR, SLL
200   FORMAT (2I0,F0.0)
      WRITE (4,300) N, NBAR, SLL
300   FORMAT (' TAYLOR SYNTHESIS - SHELTON' / ' N=',I5,
     1   2X,'NBAR=',I5,2X,'SIDELOBE=', F5.2)
      AL2 = 0.30102999566398
      ALE = 0.43429448190325
      PI = 3.14159265358979
      M = (N-1)/2 + 0.1
      IE = 1
      IF (N .EQ. (2*M+1)) IE = 0
      A = (SLL + 20.0*AL2) / (20.0*PI*ALE)
      XN = FLOAT (N)
      XN12 = FLOAT (NBAR) - 0.5
      N1 = NBAR - 1
      ALPHA = SQRT (A*A + XN12 * XN12)
      DO 1 I=1,N1
       XI12 = FLOAT (I) - 0.5
       BETA = SQRT (A*A + XI12 * XI12)
       Z(I) = ((2.0*PI/XN)/ALPHA) * FLOAT (NBAR) * BETA
1     CONTINUE
      DO 2 I=NBAR,M,1
       Z(I) = ( FLOAT (I) * 2.0*PI)/XN
2     CONTINUE
      EO = 1.0
      DO 3 I=1,M
3     EO = EO * (1.0-COS(Z(I)))
      DO 5 I=1,N1
       AM(I) = 1.0
       DELTA = (2.0*PI * FLOAT(I))/XN
       IF (IE .EQ. 1) AM(I) = COS (DELTA/2.0)
       DO 4 J=1,M
4      AM(I) = AM(I) * (COS(DELTA) - COS(Z(J)))
       AM(I) = AM(I)/EO
5     CONTINUE
      DO 6 I=1,M+1
       XI = 2*I - 2
       IF (IE .EQ. 1) XI = XI + 1
       EN(I) = 0.0
       DO 7 J=1,N1
        XJ = FLOAT (J)
        EN(I) = AM(J) * COS((PI*XI*XJ)/XN) + EN(I)
7      CONTINUE
       EN(I) = 2.0 * EN(I) + 1.0
6     CONTINUE
      DO 50 K=1,M+1
```

```
       L = N+1-K
       EX(K) = EN(M+2-K)/EN(M+1)
 50   EX(L) = EX(K)
      WRITE (4,301)
      WRITE (4,55) (I,EX(I), I=1,N)
 301  FORMAT (' ELEM. NO.', 3X, 'EXCITATION')
 55   FORMAT (5X,I2,5X,F14.7)
      STOP
      END
```

```
      C                    PROGRAM CONTAYL
      C*******
      C*  LINEAR       ARRAY SYNTHESIS USING CONVOLUTION METHOD.
      C*  TAYLOR SIDELOBE DESIGN
      C*  ODD NUMBER OF ELEMENTS.
      C*******
            REAL PSI(100)
            REAL C(100),AA(100),A1(100),A2(100),A3(100),CONV(100)
            DATA A1/100*1./, A3/100*1./, CONV/100*0./, AA/100*0./
      C*******
      C*  N = NUMBER OF ELEMENTS IN THE ARRAY. MUST BE ODD!!
      C*  SLL = SIDE LOBE LEVEL IN DBS.
      C*******
            PI = 3.1415297
            WRITE (1,100)
      100   FORMAT (' ENTER DATA: N,NBAR,SLL')
            READ (1,200) N,NBAR,SLL
      200   FORMAT (2I0,F0.0)
            M = N-1
            NN = (N+1)/2
            MM = (N-1)/2
            MD = (MM/2) + 1
      10    CALL TAYLX(MM,SLL,PI,PSI,N,NN,M,NBAR)
      C********
      C*  CREATE THREE ELEMENT ARRAYS
      C********
      30    DO 40 I = 1,MM
             A2(I)  =  -2. * COS(PSI(I))
      40    CONTINUE
      C********
      C*  REPEATED CONVOLUTION OF 3-ELEMENT ARRAYS
      C********
            CONV(1) = A1(1)
            CONV(2) = A2(1)
            CONV(3) = A3(1)
            L = 1
            K = 5
            LX = 0
      50    LL = NN-L
            LX = LX+1
      60    L = LL
            C(1) = A1(L)
            C(2) = A2(L)
            C(3) = A3(L)
            DO 70 I = 1,3
             DO 70 J = I,K
              JJ = J-I+1
              AA(J) = AA(J) + CONV(JJ)*C(I)
      70    CONTINUE
            DO 80 I = 1,K
             CONV(I) = AA(I)
             AA(I) = 0.0
      80    CONTINUE
            K = K+2
```

```
      IF (L.EQ.MD) GO TO 90
      LL = NN+1-L
      LSUM = L+LX
      IF (LSUM.EQ.NN) GO TO 60
      GO TO 50
90    CONTINUE
C     WRITE (4,401)
C 401 FORMAT ( '         PSI ZEROS'/)
C     DO 110 I = 1,M
C      WRITE (4,500) I,PSI(I)
C 110 CONTINUE
500   FORMAT (10X,I2,10X,F10.6)
      WRITE (4,600)
600   FORMAT ( '         CURRENTS'/)
      DO 120 I = 1,N
       WRITE (4,700) I,CONV(I)
120   CONTINUE
700   FORMAT (10X,I2,10X,F10.6)
      STOP
      END
      FUNCTION COSH(R)
      Y = EXP(R)
      COSH = (Y + (1.0/Y))/2.
      RETURN
      END
C********
C*  INVERSE HYPERBOLIC COSINE FUNCTION
C********
      FUNCTION RCOSH(R)
      RCOSH = ALOG(R + SQRT(R*R - 1.0))
      RETURN
      END
C********
C*  COMPUTATION OF TAYLOR ZEROS
C********
      SUBROUTINE TAYLX(MM,SLL,PI,ZEROS,N,NN,M,NBAR)
      REAL ZERO(50), ZEROS(100), MEMA,MEMB
      NBAR1 = NBAR-1
      A = (SLL + 6.0202)/27.2875
C********
C* COMPUTE ZEROS FROM 1 TO NBAR
C********
      DO 10 I = 1,NBAR1
       RI = I
       MEMA = (A*A) + ((RI-.5)**2.)
       MEMB = (A*A) + ((NBAR-.5)**2.)
       ZERO(I)=(((2.*PI)*NBAR)/N)*((SQRT(MEMA))/(SQRT(MEMB)))
10    CONTINUE
C********
C*  COMPUTE ZEROS FROM NBAR TO M
C********
      DO 20 I = NBAR,MM
       RI = I
       ZERO(I) = (2*PI*RI)/N
```

```
20    CONTINUE
C     WRITE (4,100)
C 100   FORMAT ( '                 ZEROS')
C       DO 30 I = 1,MM
C           WRITE (4,200) I,ZERO(I)
C  30   CONTINUE
C 200   FORMAT (10X,I2,10X,F10.6)
      DO 40 J = 1,NN
       ZEROS(NN-1+J) = ZERO(J)
       ZEROS(NN-J) = ZERO(J)
40    CONTINUE
      RETURN
      END
```

ED
8